



NEPALESE SOCIETY OF
STUDENT RESEARCHERS



5-Day Online Workshop on

“MATLAB FOR RESEARCHERS”

Symbolic Math and Simulink Basics



08.00 PM
'till end



6–10, September
2025

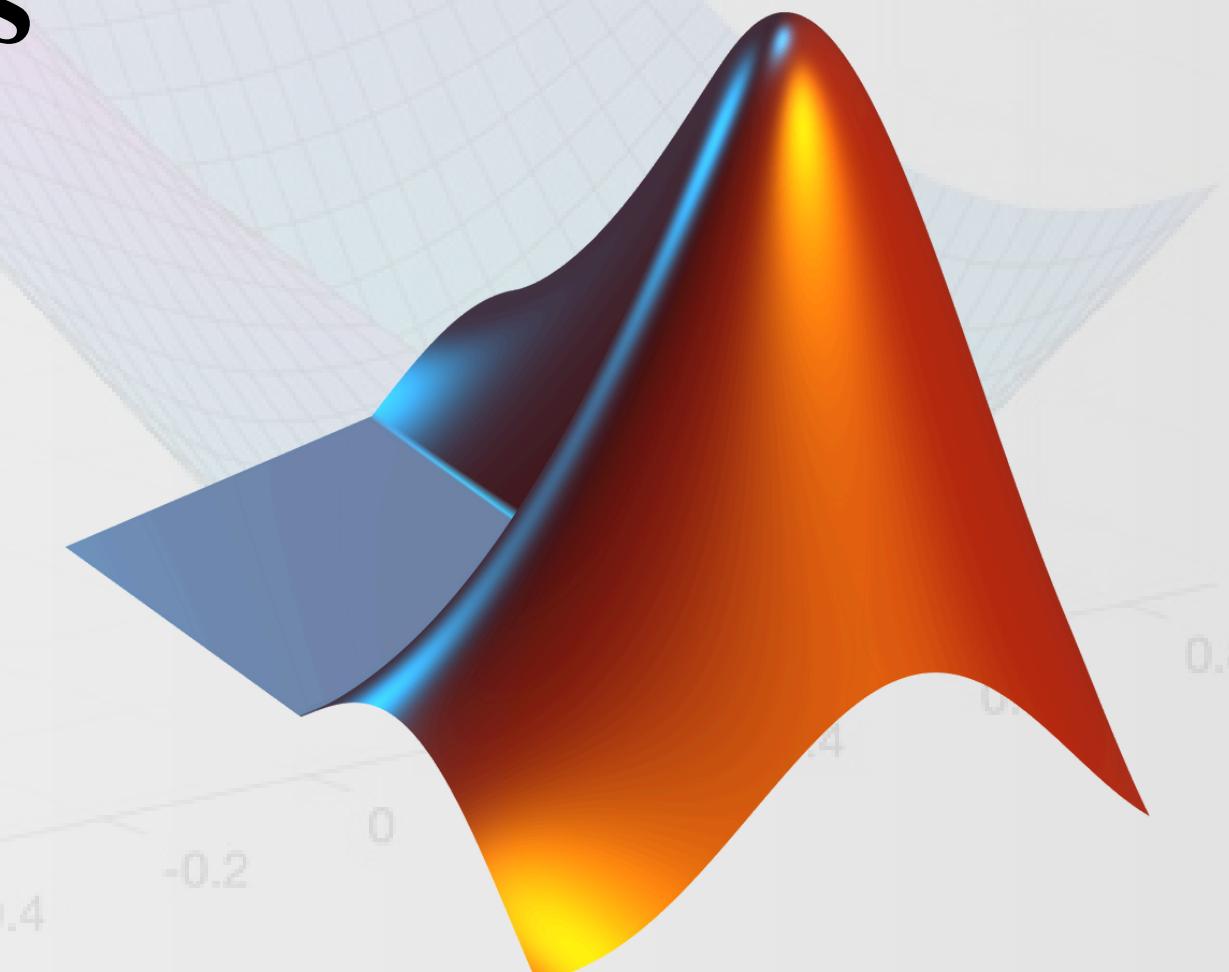


Microsoft
Teams

Janak Singh Dhami

Founder & President

NSSR Nepal, Tri-Chandra Research Group



Workshop Overview

1

MATLAB Fundamentals for Research

Learn MATLAB environment, variables, arrays, matrices, scripts, functions, and control structures with hands-on experimental data processing practice.

2

Data Handling & Visualization

Import/export datasets, clean and analyze data, apply basic statistics, and create professional 2D/3D plots with publication-ready visualizations.

3

Symbolic Math Toolbox: Algebra & Calculus

Explore symbolic variables, simplify expressions, solve equations, perform differentiation, integration, and Taylor expansions with applied research-oriented examples.

4

Symbolic Differential Equations & Transforms

Solve ODEs, initial value problems, apply Laplace/Fourier transforms, and study symbolic matrices with harmonic oscillator and circuit models.

5

Simulink Basics for Researchers

Discover Simulink environment, model building, continuous/discrete systems, and simulate physical/biological models through practical block diagram exercises.

6

Final Project Day [Sept. 10]

Integrate MATLAB and Simulink skills to complete a research-style project, involving dataset analysis, symbolic computation, and dynamic simulation.



What is Programming Language?

1. **Programming** = giving instructions to a computer to solve problems.
2. Uses logic + math + structured steps.
3. Computers only understand binary (0s and 1s) → we use high-level languages.
4. **MATLAB** = high-level scientific programming language (focuses on research & data).

Example (Real-life analogy):

Recipe → steps to cook → like a program
Ingredients → raw data → like variables
Dish → final cooked food → like program output

Our Objectives of First 2 days

1. Understand MATLAB basics and environment
2. Learn variables, arrays, and matrix operations
3. Write scripts, functions, and control structures
4. Handle and clean datasets
5. Create professional visualizations
6. Prepare for advanced symbolic and simulation topics

Prerequisites for Workshop

- 
- Basic knowledge of mathematics**
 - Familiarity with programming logic**
 - Laptop with MATLAB or GNU Octave installed**
 - No specific prerequisites are needed.**

Audience

1. Bachelor's and Master's students in Science & Engineering
2. Early-stage researchers and project students
3. Professionals interested in computational data analysis

Getting Started with MATLAB

1 What is MATLAB?

2 Why MATLAB?

3 Features of MATLAB

4 Uses of MATLAB?

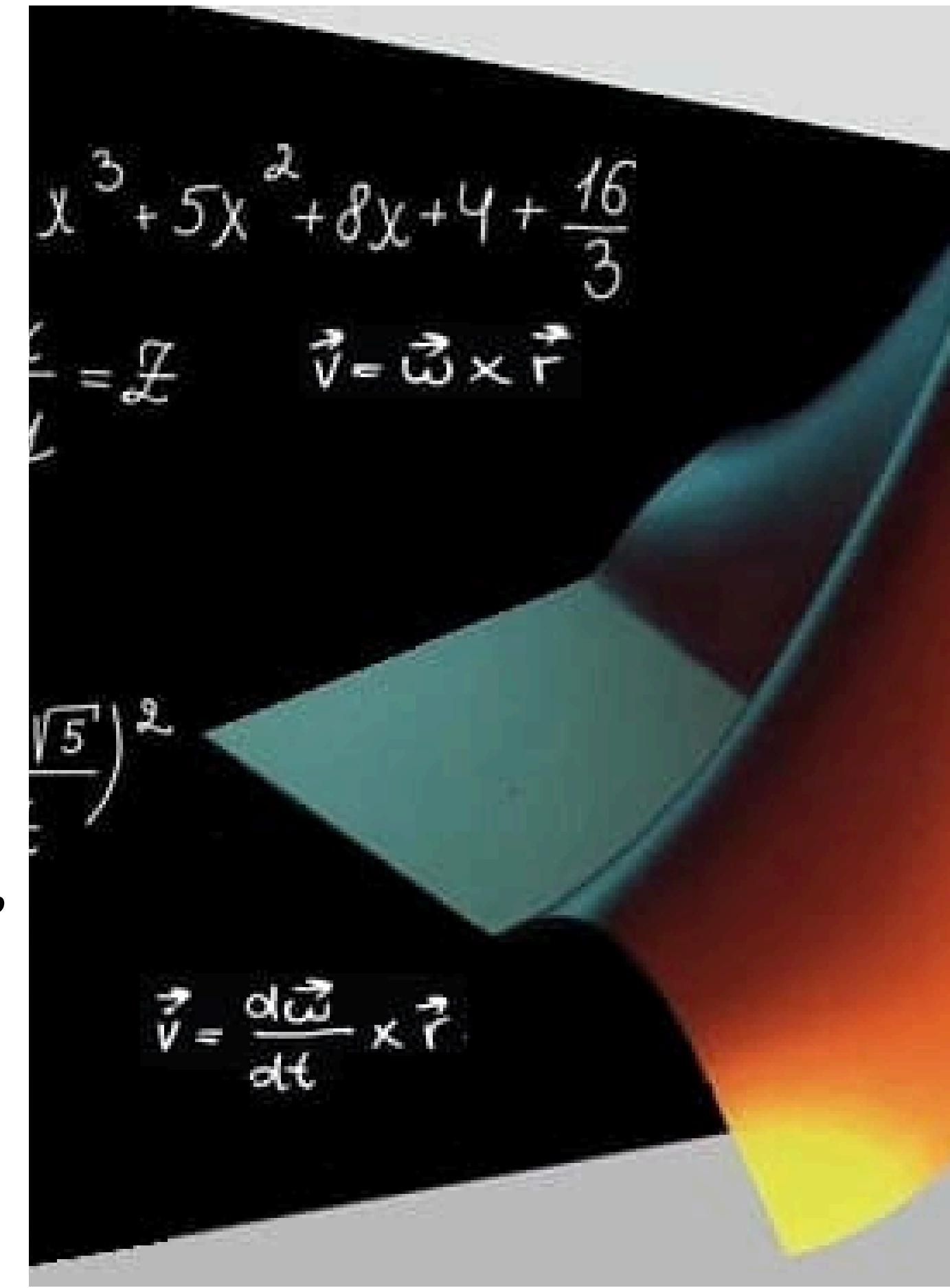
5 History of MATLAB

What is MATLAB?



What is MATLAB?

1. **MATLAB** = MATrix LABoratory → a high-level programming language designed for working with matrices, arrays, and numerical computation.
2. A computing environment + IDE that integrates programming, visualization, and analysis in one platform.
3. Developed by **MathWorks**, widely used in academia, research, and industry.
4. Specially built for scientific computing, data analysis, engineering simulations, and algorithm development.
5. Comes with toolboxes for different fields: **signal processing, image processing, machine learning, optimization, control systems, etc.**
6. Supports **visualization, scripting, symbolic math, and integration with C, Python, and hardware platforms.**

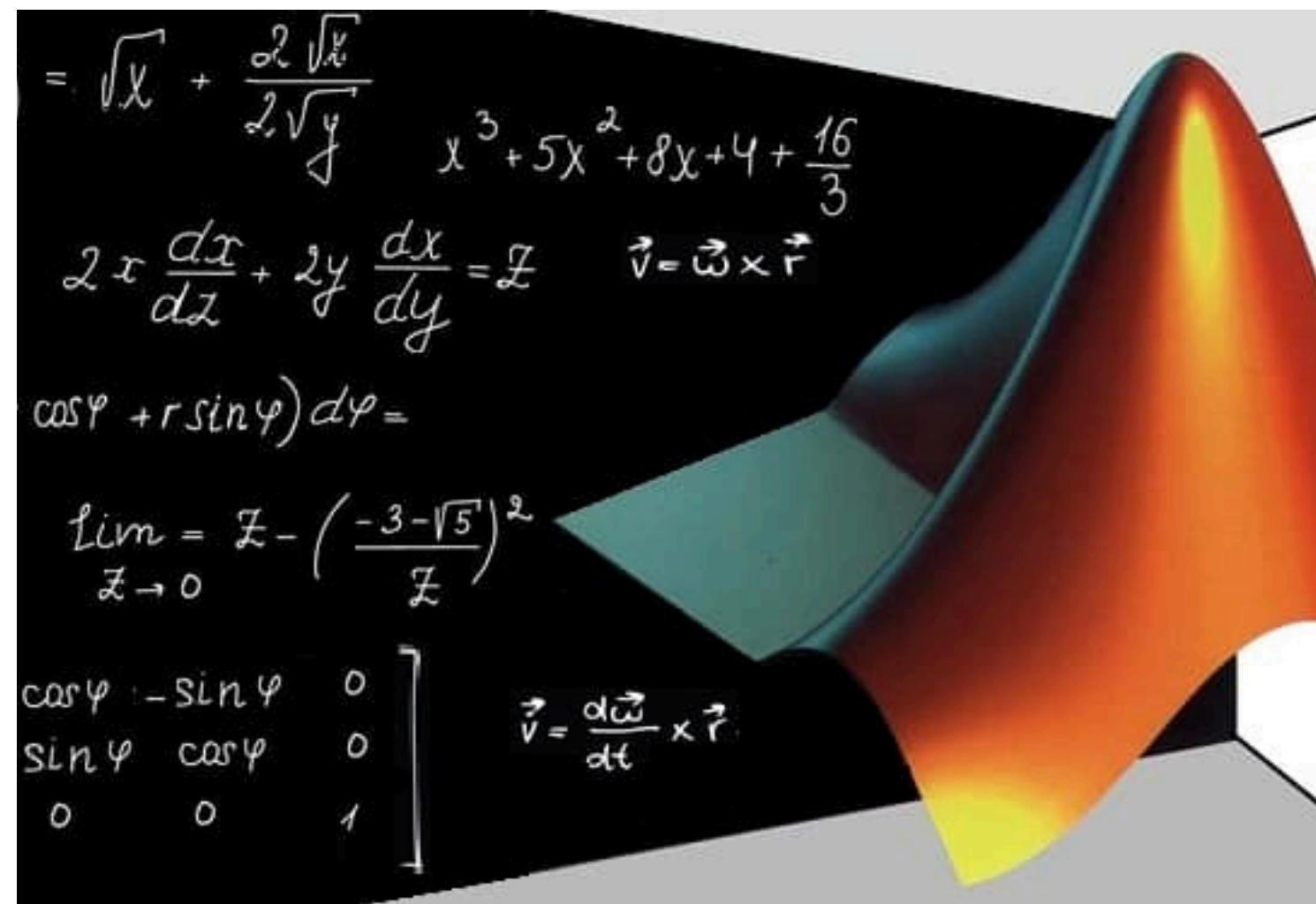


History of MATLAB?

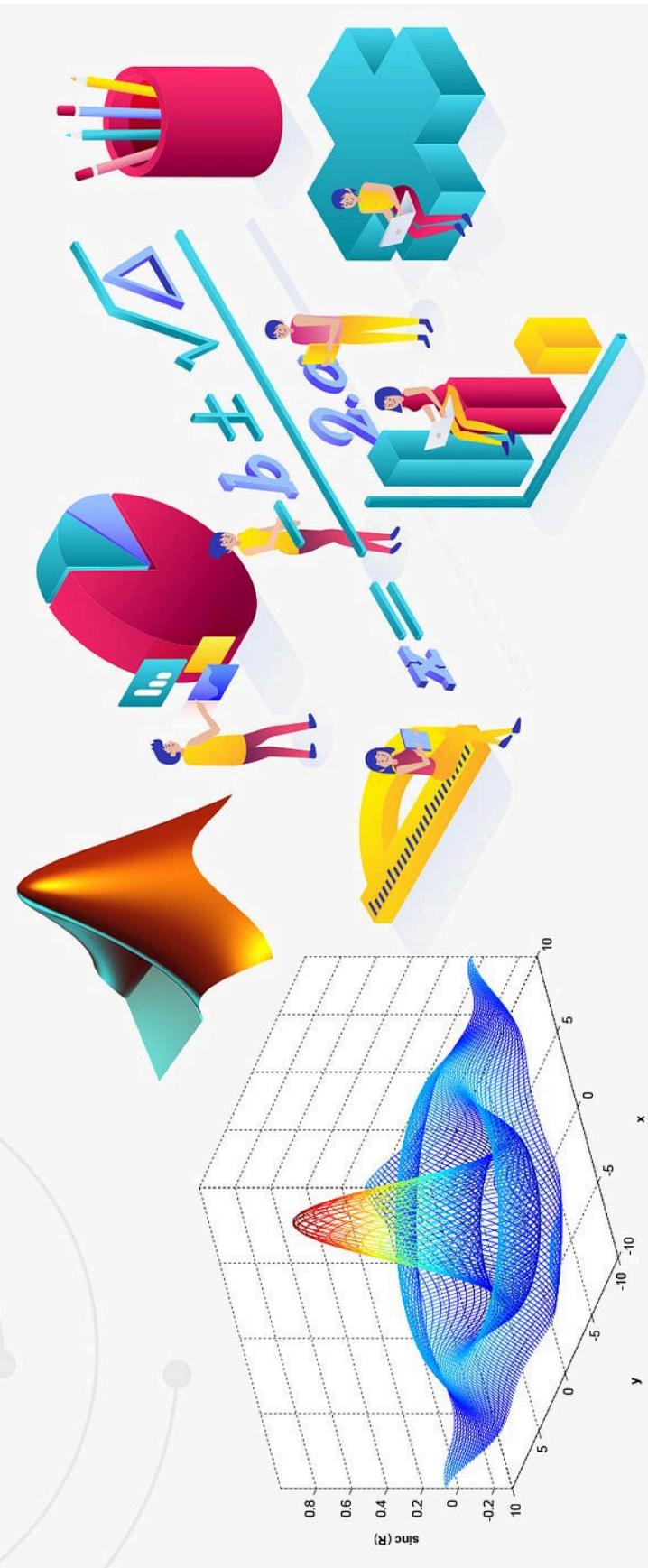
1. **1970s** – **Cleve Moler** (Professor, Mathematician) created MATLAB to help students access LINPACK & EISPACK (Fortran libraries for linear algebra).
2. Initially shared freely with students and colleagues, became popular in universities.
3. **1984** – MathWorks founded by Cleve Moler, Jack Little, and Steve Bangert; **MATLAB released commercially**.
4. Grew rapidly in Engineering, Physics, Mathematics, and Research communities due to easy matrix handling.
5. Over time, added **toolboxes**, **Simulink**, **graphics**, and **symbolic math** to expand functionality.
6. **Today**, **MATLAB** is a global standard in academia, research, and industry for data analysis, simulation, and visualization.



Why MATLAB?



1. Easy matrix/array handling
2. Powerful visualization tools
3. Extensive toolboxes
4. Academic and industry standard
5. Research-friendly (fast prototyping, reproducible results)



Uses of MATLAB?

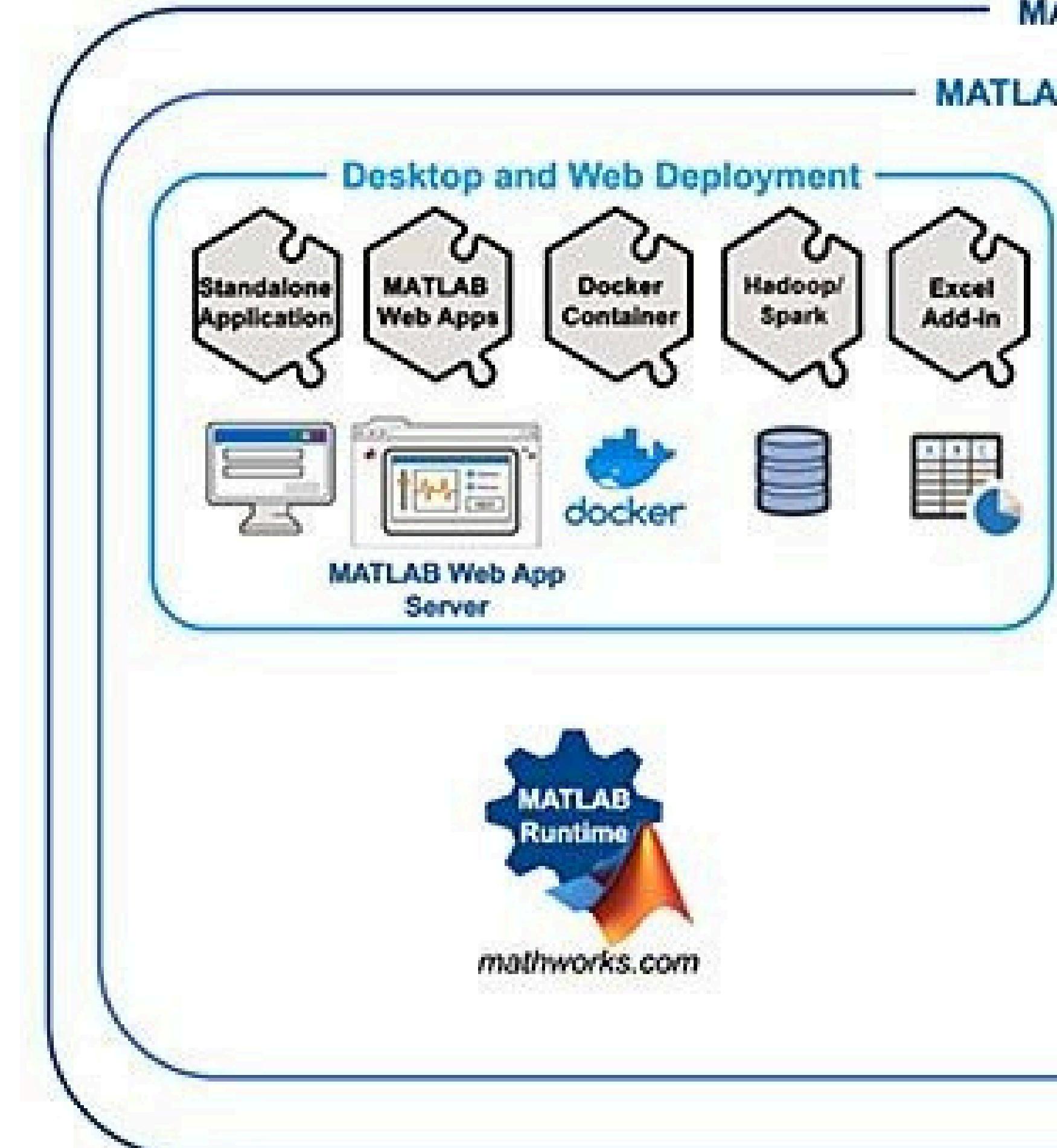
1. Scientific computing & research
2. Data analysis and visualization
3. Signal and image processing
4. Machine learning and AI
5. Simulation and modeling (Simulink)
6. Computational Biology
7. Computational Finance
8. Control systems and engineering applications

Features of MATLAB

1. It is a high-level language for numerical computation, visualization and application development.
2. It also provides an interactive environment for iterative exploration, design and problem solving.
3. It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.
4. It provides built-in graphics for visualizing data and tools for creating custom plots.
5. MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.
6. It provides tools for building applications with custom graphical interfaces.
7. It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

Components of **MATLAB**

- 1** Command Window
- 2** Current Folder
- 3** Editor/Live Editor
- 4** Workspace
- 5** Command History



MATLAB Environment

1. Command-based and script-based execution
2. Integrated development environment (IDE)
3. Flexible file handling (.m, .mat, .mlx)

Live Editor

1. Combines code, output, text, and figures in one notebook-style document
2. Great for teaching, documentation, and reproducibility

Workspace

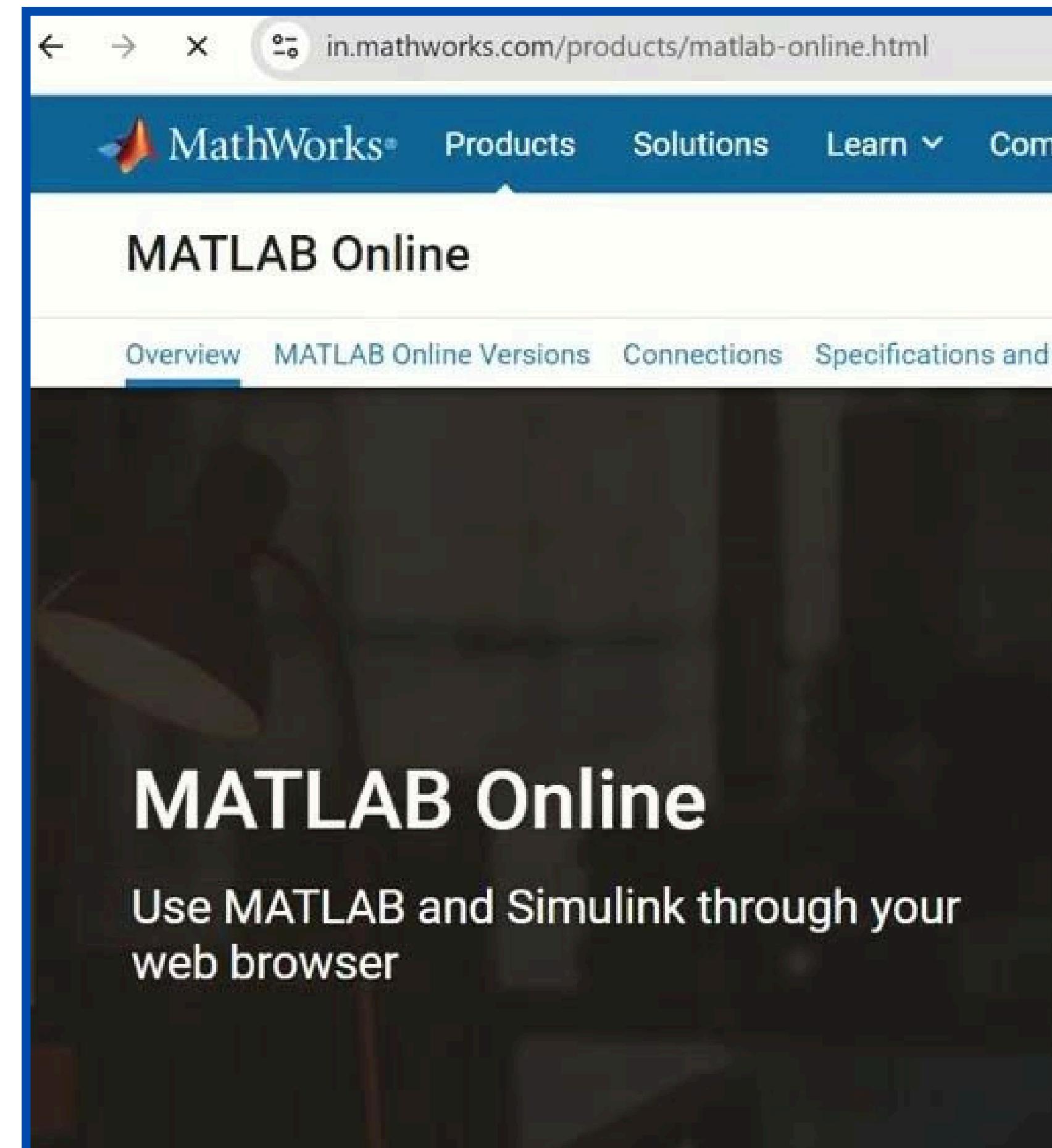
1. Stores variables during MATLAB session
2. Check variables with whos
3. Clear memory: clear
4. Save/load workspace: save, load

Syntax

1. Case-sensitive language
2. Commands end with ; to suppress output
3. Vectors: [1 2 3]
4. Ranges: 1:0.5:5

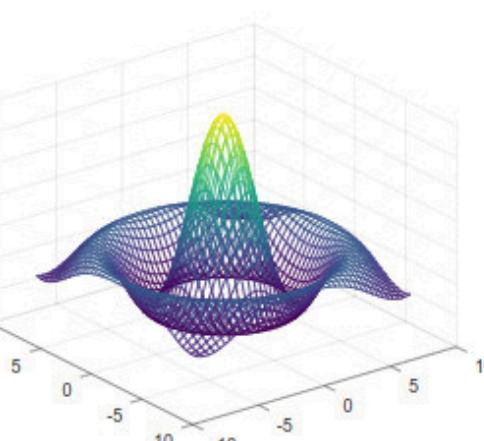
MATLAB Online & Scripts

1. **MATLAB Online:** browser-based (no installation)
2. **Scripts:** .m files containing series of commands
3. Run using Run button or command window



GNU Octave (Alternative)

1. Open-source, high-level programming language for numerical computation.
2. Mostly compatible with MATLAB syntax, ideal free alternative for students and researchers.
3. Designed for matrix operations, linear algebra, data analysis, and plotting.
4. **Free and cross-platform:** runs on Windows, macOS, and Linux.
5. Supports scripts, functions, loops, conditionals, and plotting like MATLAB.
6. **Limitations:** lacks some proprietary MATLAB toolboxes and advanced Simulink features.
7. **GUI & plotting:** command-line interface + basic GUI; supports 2D/3D visualization.
8. **Research & teaching:** widely used in courses, projects, and open-source research.
9. **Getting started:** install from gnu.org/software/octave, run .m scripts, use standard MATLAB commands.



 **GNU Octave**

```
b = [4; 9; 2] # Column vector
A = [ 3 4 5;
      1 3 1;
      3 5 9 ]
x = A \ b      # Solve the system Ax = b
```



Let's Start
Hands-on.





THANK YOU!

Hope you enjoyed learning MATLAB fundamentals and hands-on exercises.

Janak Singh Dhami

info@jsdhami.com.np

Founder & President, Nepalese Society of Student Researchers - NSSR



SEE YOU TOMORROW FOR DAY 2: DATA HANDLING & VISUALIZATION.